MASTER'S THESIS

Detecting concept drift in real-life event logs and using process variant analyses for examining the found drift

Claessen, R.H.M. (Roel)

Award date:
2020

Detecting concept drift in real-life event logs and using process variant analyses for examining the found drift

University:	Open University, Faculty of Management, Science & Technology Master's program - Business Process Management & IT
Degree program:	Open University of the Netherlands, Faculty of Management, Science & Technology Business Process Management & IT master's program
Course:	IM0602 BPMIT Graduation Assignment Preparation IM9806 Business Process Management and IT Graduation Assignment
Student:	Roel (R.H.M.) Claessen
Identification number:	
Date:	21-06-2020
Thesis supervisor:	Dr. Lloyd (L.W.) Rutledge
Second reader:	Prof. dr. ir. Stef (S.M.M.) Joosten
Version number:	1.0
Status:	Final version

Abstract

We present an approach to detect concept drift in a real-life event-log and examine the found drift via variant analysis. The approach can help trace best practices in given time periods or departments. It can be utilized by anyone who wants to get a deeper understanding of their process and see where improvement is possible.

Unforeseen changes that happens over time in processes can occur through changing situations, for instance, amending regulation or seasonal effects. The phenomenon of unforeseen change over time is defined in the literature as concept drift.

There is research done on concept drift in the process mining community and solutions have been provided. We present a practical and business friendly approach to detect and examine concept drift, partly based on an existing solution. One solution in the form of a ProM plug-in is evaluated on a real-life event log which covers a considerable period. Furthermore, an unexpected applicability of the plug-in to demonstrate volatility in a process is presented. Moreover, we investigate how and to what extent found concept drift and volatility can be examined using variant analyses. Next to this we elaborate which shortcomings apply when using a real-life event log, especially when a profound business context is missing.

Key Terms

Business Rule Mining, Process Mining, Concept Drift, Variant analyses, Volatility

Contents

Tables

Figures

1. Introduction

Customers are becoming increasingly demanding and legislation is constantly being amended. For businesses to keep a competitive advantage it is necessary to have efficient processes that can operate at low costs. Studying how processes behave and how they can be improved is being studied in the context of Business Process Management.

In this thesis we treat concept drift in the Business Process Management context. In this chapter we introduce and explore the topic, indicate the relevance of this thesis, and determine the problem statement and end with the main line of approach for the further thesis.

1.1. Exploration of the topic

Business process management and business rule management both study the management and execution of tasks [1]. However, they do so from different perspectives. Business process management takes an activity and resources viewpoint, while business rules management approaches tasks from a guideline or knowledge viewpoint [2].

Business rules are everywhere. Everything in an organization is governed by rules [3]. Business rules are part of business processes and tell the organization what to do at atomic level, meaning that they cannot be further broken down [4]. Setting business rules is essential for every organization to turn strategy into actions. Business rules can be explicit, which means documented or implicit which behave in an organic manner. Especially the implicit rules are hard to manage and enforce. Explicit rules can be of low quality and thus interpretable in diverse ways, which makes them hard to enforce. It seems that collecting and recording business rules can be a costly and painstaking job. Process mining can be -among other benefits- the answer to improve this process.

Especially in this time of organizations regarding themselves as data-driven, a great quantity of data is stored and readily available. There are already process mining techniques developed that discover, analyze, and enhance process models using event logs. This creates the opportunity to mine data for business rules, which are more fundamental. This can offer organizations the chance to improve and control the business rules if possible, in an automated way.

Over time business processes or business rules change. This can happen because processes or methods are formally changed, but they can also change over time without specific new instructions. This thesis addresses the latter. Unforeseen changes over time within processes or business rules, known in literature as concept drift.

1.2.Relevance

The goal of the Business Rule Mining Masters Circle is to develop process mining techniques to derive business rules from event logs. More specific, the purpose of this thesis is to develop techniques to detect concept drift in processes and examining them through process variant analysis. Concept drift is an unforeseen change, in this case in a process, where the point of change is not necessarily known. For example, when in a bank too much work is on hands this could have consequents for the number of applications that are processed. These changes should be visible in the event logs while the formal process is not necessarily changed. Analyzing such changes is important when supporting or improving operational processes. In the data mining community, such second-order dynamics are introduced as concept drift by [5]. In their article they present three main problems regarding concept drift:

1. Change detection: This the most basic problem concerning concept drift, determine that a concept drift has taken place. After this it is important to set the period in which this has happened.
2. Change Localization and Characterization: After de concept drift has been detected it is necessary to determine the reason for the change in the process. This is a challenging task where both the identification of the change perspective and the exact change is involved.
3. Unravel Process Evolution: When the change is identified, localized, and characterized it needs to be put in perspective. The article states that there is a need for techniques and tools that can utilize these discoveries. When the evolution of a process can be deciphered it should lead to the discovery of the change process.

Other research [6] has demonstrated that process discovery algorithms can behave poorly when event logs contain drifts. It can happen that casual relations between events appear of disappear and therefore cannot be resolved. Further research is therefore desirable.

1.3.Problem statement

In the before subsections we have concluded that from a business context it is interesting to detect and analyze unforeseen changes -in other words concept drift- in a process. We are curious to research that if after the concept drift has been detected we can explain this unforeseen change through variant analysis. The problem statement is therefore:

How and to what extent can process variant analyses be used to examine concept drift from a control flow perspective?

1.4.Terms of reference

We want to add to the research domain the validation of existing concept drift techniques performed on a real-life event log. Hereby we want to focus on not only detecting concept drift but also explaining it by doing a process variant analysis.

For this the following questions with regards to the problem need to be answered:
1. *How can concept drift be detected in event logs?*
2. *How can variant analyses help in examining changes in the process?*
3. *What tools and models are available to support this analysis?*

By performing an extent literature review we expect to answer the above questions.

1.5.Main lines of approach

In the before subsections we have introduced the topic and relevance of the thesis. Furthermore, the main problem statement plus follow up questions are presented.
In chapter 2 we discusses the theoretical framework meaning we present the research approach, an extensive presentation of the literature review and its conclusions. Chapter 2 ends with the objective for follow-up research. Chapter 3 is all about the methodology, how the data is gathered and analyzed and concludes with the plan of approach. Chapter 0 focuses on the experiments and the treatment of the techniques. This chapter is followed by chapter 5 which presents the discussion which consist of the conclusions, practical recommendations, guidance for future work and lastly a reflection on this work.

2. Theoretical framework

The topic of the master's thesis is handed by the thesis supervisor to our master's circle. A broad subject, business rule mining, is allocated to us. To gain a better understanding on this subject and the current field of knowledge about business rule mining and subjects around it are revealed. Extra attention is paid to theories and ideas about the specific subject this thesis engages, namely concept drift and variant analyses.

2.1.Research approach

To frame the research a thorough research approach is executed. The starting point of the literature review is process mining which is being researched for quite some time and a lot of relevant scientific articles and books can be consulted. Relevant business rule mining articles are more difficult to find and are available in lesser numbers.

The chosen approach for finding suitable literature in this thesis consist of three parts. First the articles provided by the thesis supervisor on which the topic is determined are studied. Secondly search quotes are used in the OU library and Google Scholar for relevant literature. Searches have been conducted using key words as, "Process Mining", "Business Rule Mining", "detecting Business Rules", Process Mining & Finance", "Concept Drift & Process Mining", "Variant analysis" and "ProM". This list is not exhaustive, and the key words are used in different compositions. Third and last, the citations and references included in the found articles are scoured for more relevant literature.

Based on this literature review, the articles are divided into three subjects. 1.) Process mining because this is a field that has been the subject of plenty research. This can help to put business rule mining in perspective, to offer general ideas and direction. 2.) Variant analysis in processes, to see which research is being conducted and where there is need for future work. 3.) Concept drift in process mining, this because in different found articles future work is hinted in this direction.

2.2.Execution of the literature review

In the end, around 30 articles -in a greater or lesser extent- were viewed, read, and studied. Finally, six main scientific articles are selected to derive the research goal and the sub-goals of this thesis.

Bolt et al. [7] addresses problems with analyzing and comparing different variants of the same process. It states that traditional process mining techniques assume that processes demonstrate homogenous behavior, and thus can easily be compared. The reality is more unruly: it seems that business processes are dynamic by nature. It states that existing approaches focus on the control flow perspective and only detect differences that are not statistically significant. Additionally, a

technique is proposed based on transaction systems that detect statistically significant differences between process variants. It reveals the similarities and differences of the business rules between them, using event logs as input. The technique can pinpoint differences that previous approaches failed to provide.

Polpinij J et al. [8] defines a business process as consisting of a set of logically related activities, performed by their relevant roles or collaborators, to intentionally achieve the common business goals [9]. They state that they have found significant hidden knowledge in the business process model repositories. They consider the sequential patterns as business rules. With their work they support organizations in harvesting business rules from their existing business process models. In the conclusion they indicate that the first implementation of business rules in business processes are adequate. But the ever-changing environment ensures that organizations constantly must reconsider business activities to remain competitive and to keep a lasting business model. The presented future work is described very broadly. Investigating how to extract other useful information from repositories or business process models.

Carmona et al. [6] states that little work regarding concept drift in process mining is done. The article presents online mechanisms for detecting and managing concept drift. Methods are presented that can detect abrupt changes quickly and accurately. They advise future work to develop novel techniques, particularly for detecting different forms of change, particularly, gradual, long term changes.

Maaradji A et al. [10] indicates that business processes evolve through various factors, such as changing regulatory environment, competition, demand, technology capabilities and seasonal factors. Business processes tend to deal with continuous and unexpected changes. The article proposes a fully automated method for business process concept drift detection. The proposed method can accurately discover typical process changes. Future research is recommended in better understanding the process changes and pinning a detected drift. In summary they advise that in future research a method is explored where the user receives more insight and understanding in the detected drifts.

Bose R.P. et al [5] [11] provide tools for handling concept drift in process mining. They state that very few information systems provide change logs. That is the reason they focus on concept drift in process mining with only event logs as input. So not using available recorded change logs but the organic and undocumented changes of processes. Furthermore, it is stated that work has been done on concept drift in the domains of data mining and machine learning. However, the problem of concept drift has not yet been studied in the process mining community. Experience from the data mining and machine learning domain can be used but new techniques are needed for detecting concept drift in process models given the complexity and the specific nature of the change.

In [5] and [11] three perspectives or classes regarding business processes that can be subject to change are specified: control flow, data, and resource perspective. One or more of classes can be subject to change at any given time.

Control flow & Behavioral Perspective: This class is about behavioral or structural changes in the process model. A structural change can be an insertion, deletion, substitution and reordering of a portion of the process. In other cases, the change is through behavior. This is the case when the structure of the flow stays unchanged, but the interpretation of a process step changes.
Data perspective: The perspective refers to changes in the handling of data during the process. This can be the requirement, usage of generation of data in the process. A certain task produces or requires data or information.
Resource Perspective: This concerns changes in resources, their roles, and the organizational structure. This in the perspective of their influence on the implementation of a certain process. For example, if a certain raw material is not available, certain parts of the process can no longer be conducted, or they must be conducted in an adapted way.

Changes can be momentary or permanent. Momentary changes obviously do not last long and only effect a limited number of cases. Permanent changes however are more persevering and due to last longer. Changes that are perceived to induce a drift in the concept, or process behavior. Bose R.P. et al. identify four types of concept drift:

Sudden Drift (a): In this kind of drift a process is replaced for a new process as illustrated in figure 1 (a). As soon as the new process is followed, the old process is no longer in use. This is common in processes that follow strict regulations or legislation. Once a rule or law has been amended, the old process is obsolete.
Recurring Drift (b): Some processes emerge again after a given time as seen in figure 1 (b), then sometime later the old process is reinstalled. This form of drift occurs for instance in processes that are influenced by seasons.
Gradual Drift (c): In this case, as by sudden drift, a new process is followed. This with the difference that here the processes can run simultaneously. The old process is less and ultimately no longer applied. This can occur for instance when the new process applies to all new orders. Existing orders are still processed via the old process. If there are old orders, the old process is still active. At the last processing of an old order, only the new process will be active.
Incremental Drift (d): The last form of drift is when a when a process is subsequently and successively changed by a new process. These are small and incremental changes as seen in figure one (d). This can be seen in modern way of working like rapid application development, in other words agile way of working.

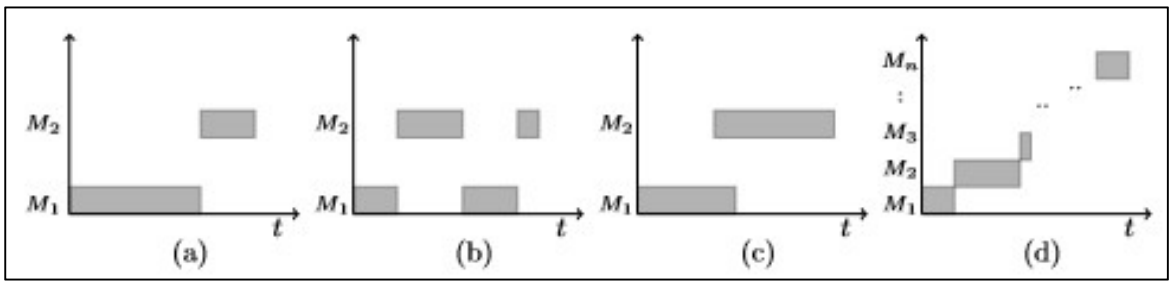

Figure 1: Different types of drift - X = Time Y = Process Variants - Bose et al: Handling Concept Drift in Process Mining

2.3.Conclusion from literature review

The goal of the above literature review is to find answers to the questions stated in section 1.4 which were defined for the high-level problem statement:

How and to what extent can process variant analyses be used to examine concept drift from a control flow perspective?

We elaborate on the three questions below:

1. *How can concept drift be detected in event logs?*

There are different techniques presented in the literature to detect concept drift in event logs. For one technique a plug-in has been developed in ProM based on a non-real-life event log. There is a need for more research to validate this technique based on real-life event logs, preferably event logs that span a longer period. Furthermore, there are techniques required that can offer more insight and understanding on the detected concept drift.

2. *How can variant analyses help in examining changes in the process?*

Out of the literature review can be concluded that is possible to examine changes with variant analyses in process. We have not found research that utilize variant analysis on detected concept drift points.

3. *What tools and models are available to support this analysis?*

The main tool used in the scientific process mining community being is the ProM architecture. ProM is very suitable for researchers and developers because it is open source and they can develop plug-ins themselves. Relevant for this thesis is that there is a concept drift plug-in available. See for a more extensive overview for the tools used in this thesis subsection 3.2.

Bases on the above the following can be concluded. A solution to detect changes in event logs for sudden drifts from a control-flow angle has been designed, but further validation of this technique, especially with real-life event logs is desirable. Although there is relevant literature on variant analyses which examine changes in processes, we could not find articles in which variant analyses is used to examine and explain drift points. Thus, we think there is an interesting and relevant gap in the literature. Especially because multiple articles mention that further research regarding the explanation of concept drift is desired. We think it is relevant to validate the concept drift technique on a real-life event log. To see what challenges, we come across in managing a real-life event log and see if we can provide general solutions to overcome these challenges. Next to this we are curious if and to what extent variant analysis is suitable for examining the found concept drift.

2.4.Objective of the follow-up research

Primarily, our literature review makes it clear that processes are not static and are constantly subject to change. So, this thesis is about -the examining of- unforeseen change. The change can have various reasons for instance new or changing regulations, technological developments, or the behavior of the competition. Changes can be expected or unexpected. In this thesis, we are interested in detecting the unexpected changes, concept drift, in a real-life event log and see if and to what extent we can explain these changes via variant analyses.

Therefore, we present the following follow-up research:

1. Validating an existing technique, the Process Drift Plug-in in ProM [5] for detecting concept drift using a real-life event log extending over at least 3 years.
2. After performing the activities in point 1 we want to research how and to what extend variant analyses can help in explaining the detected concept drift.

3. Methodology

With this thesis we want to add value in the business process mining domain. As said in subsection 2.4, the thesis is aimed at performing at validating the concept drift plug-in on real-life event log and examining the results with process variant analyses. As listed in chapter 2 research has been done regarding concept drift detection and variant analysis. In this thesis we want to bring these techniques together. For detecting sudden drifts from a control flow perspective, successful methods have been developed. The studied articles have provided enough guidance on where future work is to be done in this specific but interesting subject.

3.1.Research method

The method for the follow-up research is are observed based experiments where the results from the detected concept drift in a real-life event log are examined trough variant analyses. The answering of the research questions will be done by performing and describing experiments and comparing the results. There is a shortcoming in the setup explained in subsection 2.4. If there is no concept drift found in the execution of point 1 mentioned performing point 2 is of little use. This means that finding a suitable event log is essential.

We want to perform the research from a management perspective. Thus, we aim to present easy to perform experiments that can be reproduced by technical and non-technical readers. To be in line with the above a mathematical, statistical, or programming background is not required to execute the experiments. A drawback of this way of working if that we are limited in using only the available models and tools.

3.2.Tools and standards

The intention being presenting a solution that can be reproduced by a non-technical performer which has a basic mathematical background. A management or business solution so to say. This means we use software and solutions of which at least a free student or trial version is available. The following programs are used in this thesis:

- Excel for preparing and understanding the event log.
- Disco [12] for exploring the process and performing experiments.
- ProM 6.9 [13], because its variety of plug-ins and possibilities. In particular:
  - the concept drift plug-in.
  - the CSV to XES plug-in.

ProM has been developed over the years and is an important framework within the in the process mining community. It is an open source framework which is extensible, it supports a large array of plug-ins used for research. Other users can use and evaluate these plug-ins.

The specific plug-in we want to validate is the Concept Drift Plug-in presented in the paper of [5]. It is based on a feature called J-measure which can be used to recognize drift in a control flow over time. In short, the J-measure makes use of variants of follow and precedes relations within a trace to determine the significance of relationship between those traces. This thesis has not the purpose of explaining the exact functioning of the J-measure. For further reading see Smyth et al. [14].

Regarding the detection of drift points, we have limited ourselves to the Kolmogorov-Smirnov test in the concept drift plug-in. The Kolmogorov-Smirnov test answers the hypothesis if two independent samples are sensitive to any kind of distributional differences and expresses this in a p-value.

Multiple other open source process mining tools are available. In addition, there are enough commercial vendors that offer tools for a fee. Above tools were chosen because they are equipped for performing the experiments and are mentioned in articles studied.

3.3. Data collection

As said in subsection 2.4 using a real-life event log is essential for validating the concept drift plug-in. This because we want to examine which challenges appear when managing these kinds of event logs in the context concept drift. Moreover, literature studied for this thesis states that future work for validating the detection of concept drift in real-life event logs is needed.

Therefore, we have chosen an event log from the yearly Business Process Intelligence challenge. The data is delivered by a German data company. Furthermore, data gathering has already been performed by the designers of the challenge. Because a prepared and freely available event log is used privacy risks are excluded. The data has already been cleared in that regard. Although it is real-life data, it has been adapted to such an extent that it is anonymous.

The event log covers the processing of applications direct payment for German farmers issued by the European Agricultural Guarantee Fund [15]. The process for distributing EU Agriculture funds are subject to complex regulations captured in European and national law. A farmer must apply yearly and eligibility is checked for each application again. This makes it a repeating yearly process, with minor changes, for example by changing (EU) regulation. The event log covers a usable period of 3 year, from 2015 until 2017.

There are limitations of this event log. Although a brief description comes with it, it is not possible to ask for further clarification. This means that we cannot verify concept drift with the people that perform the process. It therefore lacks a profound notion of the underlying business context. The size of the event log is also a challenge. The question is whether the equipment used can cope processing the event log. This is clearly one of the challenges when using a real-life data set for which additional research has been indicated.

Despite the shortcomings the event log meets our requirements, it is a real-world process and covers a period of 3 years.

3.4. Data analysis

After analyzing the data, the following is concluded. The event log contains 43.809 direct payment applications or cases, for further reference called traces. There are in total 2.5 million events over a three-year period. The shortest trace is 24 events long, the longest trace is 2.973 events long. On average there are 57 events per trace. The data is produced by a combination of automatic and manual work procedures. The workflow starts when an application is received and ends when a payment is authorized. In the workflow, the application goes through process steps that see if the direct payment is eligible and if the amount to be paid is correct. The process has the possibility to reopen an application. This can be done by the department but also by the applicant in a legal objection. Other applications go through inspections.

3.5. Plan of approach

We now want to present what exactly are going to do to get results. We describe this in the following subsections.

3.5.1. Preparation of the experiments

To start, we prepare the event log for further usage. This means transforming the CVS file to an XES file, so it can be used in ProM. After a short analyzation of the event log and the process in it we check the event log for data issues which can comprehend the result. Hereafter we present the challenges in managing a real-life event log and present solutions to make the event log suitable and more user friendly for the experiments.
Before we can examine the concept drift via variant analyses, we first need to detect concept drift in the event log. In this thesis we confine ourselves to sudden drift from a control flow perspective.

3.5.2. Configuring the experiments

As mentioned, we use the concept drift plug-in in ProM as presented in [5]. With the improved event log, we want to determine whether we can validate the concept drift plug-in on a real-life event log. In other words, how to what extent is it possible to detect concept drift with a real-life event log? Detecting the concept drift is an iterative process in with we alter the event log, the configuration of the concept drift plug-or both in to improve the results. First, we want to see if the plug-in detects concept drift at all. Then we want to further evaluate the plug-in and see whether we can obtain results that have business value.

3.5.3. Applying a variant analysis on the found concept drift

With the found results from subsection 3.5.2 we are going to apply a variant analyses. This to further validate the found concept drift and to see to what extent we can explain the found concept drift. We suspect that the locations where drift has been detected also have a larger variation in the traces. By comparing the observations from the concept drift with the variant analysis, we expect to enhance the results from the experiments with the concept drift plug-in.

3.6. Methodological issues

The techniques that are developed through the experiments can present a couple of results. The detected drift points can be true positive, which means that after validation it can be concluded that the technique works accordingly. The second result are false positives which a drift point is detected by the technique but in real life is not there. For this thesis our largest concern is the false negatives, it means there is a drift point in the event logs, but it has not been detected in the technique performed in one of the experiments. As for this thesis, as we use an existing concept drift detecting technique, in principle we do not investigate false negatives.

To ensure reliability, an open source but real-life event log is used. Next to this we chose to only use freely available tools. This means that the experiments can easily be reproduced providing the same results.

4. Results

The goal of performing the below experiments is to present techniques that solves the problems states in subsection 2.4, and to evaluate that it does so. We want to present solution that can be used in a general way and thus applied to another problem in the same domain. For this we use the real-life event log described in subsection 3.3.

Conducting experiments and managing the data can be a challenge. Adequate preparation is essential for useful and reliable results. This means that data preparation to create a usable and lightweight event log without influencing the structure of the traces is essential.

The results of the experiments are presented in the following three subsections. Subsection 4.1 presents a solution on how to manage and prepare a real-life event log for further usage. In subsection 4.2, we demonstrate how to detect concept drift from a control flow perspective and see how volatility changes over the years. Subsection 4.3 tries to explain the concept drift and the volatility between the different years and departments by performing a variant analysis.

4.1. Preprocessing the event log data

In this subsection we describe the transformations that were developed during the continual improvement process in the data analysis, defining and evaluating the hypothesis. In the end what is presented in this subsection is the outcome of many iterations.

First, we present a general understanding of the event log. While we go through the data, we check for minor data issues. The event log is too large and complex to manage and use in the tooling. We want to present a solution to simplify and reduce the size of the event log by grouping events in the traces. Lastly for the event log, a CSV file, to be usable in ProM the event log needs to be transformed to a XES file.

4.1.1. Understanding the event log and fixing minor data issues

In this case an overview of the different events including an explanation is available. For the used event log a description is provided for each of the document types and corresponding sub processes.

Document type (*Doctype*)	***Subprocess***	**Explanation**
Control summary	Main	A document containing the summarized results of various checks (reference alignment, department control, inspections)
Department control parcels (before 2017)	Main	A document containing the results of checks regarding the validity of parcels of a single applicant
Entitlement application	Main Objection Change	The application document for entitlements, i.e., the right to apply for direct payments, usually created once at the beginning of a new funding period
Inspection	On-Site Remote	A document containing the results of on-site or remote inspections
Parcel Document (before 2016)	Main	The document containing all parcels for which subsidies are requested
Geo Parcel Document (replaces Parcel document since 2016 and Department control parcels since 2017)	Main Declared Reported	The document containing all parcels for which subsidies are requested. From 2017, the Geo Parcel Document also replaces the Department control parcels document.
Payment application	Main Application Objection Chance	The application document for direct payments, usually each year
Reference alignment	Main	A document containing the results of aligning the parcels as stated by the applicant with known reference parcels (e.g., a cadaster)

Table 1: Explanation Document type and sub process [16]

It is to be advised to on beforehand go through the dataset and scan for unusual errors. In this event log there are minor irregularities that need to be fixed before it can be effectively use and to avoid wrong outliers. For the used event log minor data errors have been corrected. Timestamps of 2014 have been corrected to 2015. In this case correcting the timestamps is a minor assumption given the event log description indicates that the event log contains applications through the years 2015 till 2017.

4.1.2. Grouping events to reduce trace size

It is possible that the IT system used to record the events is set up in such a way that a new event is created for every action. These actions can be executed manually at any point in time through document specific tools or they can be scheduled automatically. The latter may be either explicitly stated in the log or implicitly apparent if a large number of actions is performed by the same user at around the same time, known as batch processing [16]. This without a new real event taking place. These consecutive events do not add value to the flow of the trace and make an event log unwieldy and slow. For performing heavy duty computer tasks like detecting concept drift, it can be desirable to group these events.

Here we want to present a solution to make an event log smaller and better manageable without losing the structure and flow of the traces. The presented solution groups events to lessen the traces and thus creating a collapsed event log.

Figure 2 presents an overview of the original event log used. With on the right the major statistics of the event log.

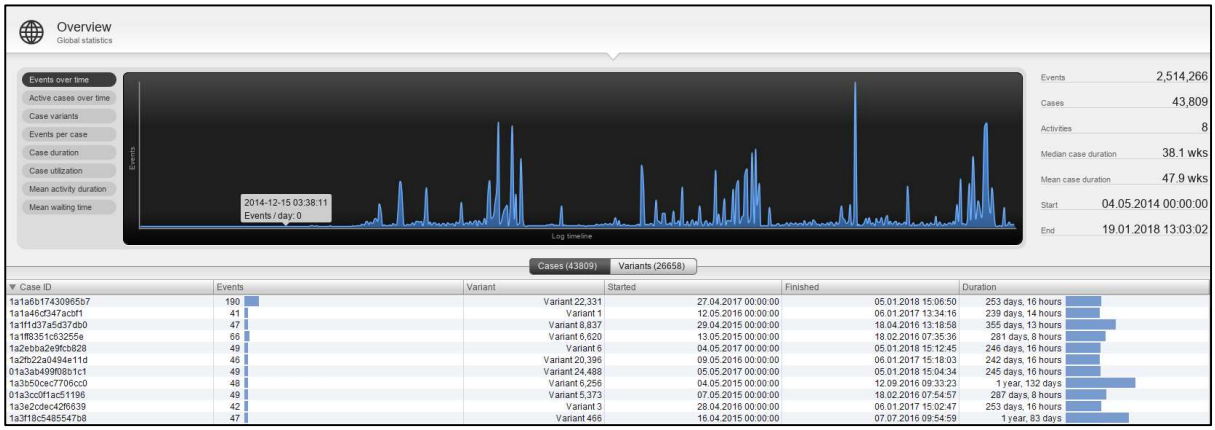

Figure 2: Overview in Disco - Original event log – See in the top right corner the statistics of the event log

As can been seen in the left part of Table 3 a trace has multiple double and consecutive values in the column *Doctype*. By grouping these double values, the size of the event log can be significantly reduced without losing the flow of the traces. For further usage of the event log, it is expected that time is an important value. We do not want to lose this value. We want to capture the total time of a set of the same events as well as the entire handling time of a trace. For this, we created four extra columns in the event log as explained in Table 2.

Column Name	Description
Start_Time	Time at which the event set started
minT	Minimum timestamp for the case (Trace started)
maxT	Maximum timestamp for the case (Trace ended)
End_Time	Time at which the event set ended

Table 2: Added columns in the collapsed event log

An example of the results of above solution is visualized in Table 3 for one trace. In short for this specific trace, eleven events are reduced to four events, this with maintaining the flow of the trace.

Case-ID	startTime	completeTime	Doctype
1a1f1d37a5d37db0	29-09-15 11:57	29-09-15 11:57	Reference alignment
1a1f1d37a5d37db0	14-10-15 13:58	14-10-15 13:58	Reference alignment
1a1f1d37a5d37db0	14-10-15 14:00	14-10-15 14:00	Parcel document
1a1f1d37a5d37db0	14-10-15 14:00	14-10-15 14:00	Parcel document
1a1f1d37a5d37db0	14-10-15 14:00	14-10-15 14:00	Parcel document
1a1f1d37a5d37db0	26-10-15 13:41	26-10-15 13:41	Entitlement application
1a1f1d37a5d37db0	31-10-15 08:03	31-10-15 08:03	Entitlement application
1a1f1d37a5d37db0	31-10-15 08:03	31-10-15 08:03	Entitlement application
1a1f1d37a5d37db0	06-11-15 11:35	06-11-15 11:35	Payment application
1a1f1d37a5d37db0	10-11-15 19:40	10-11-15 19:40	Payment application
1a1f1d37a5d37db0	10-11-15 19:40	10-11-15 19:40	Payment application

Case-ID	Start_Time	End_Time	Doctype
1a1f1d37a5d37db0	29-09-15 11:57	14-10-15 13:58	Reference alignment
1a1f1d37a5d37db0	14-10-15 14:00	14-10-15 14:00	Parcel document
1a1f1d37a5d37db0	26-10-15 13:41	31-10-15 07:03	Entitlement application
1a1f1d37a5d37db0	06-11-15 11:35	10-11-15 19:40	Payment application

Table 3: A snippet from Excel with an Example trace before and after grouping the Doctype and combining the start and end time

The changes after performing this technique in the event log are visualized in Figure 3. See at the top right that the total events have decreased significantly, this also applies to the number of events per case as illustrated in the column “Events”. As said in subsection 3.4 the longest trace in the original event log contains 2.973 events, in the collapsed event log the longest trace contains 124 events. Because of the error in the starting time in a couple events the start of the event log is now 26-01-2015 instead of 04-05-2014. This is clearly visible in the log timeline (X-axis), which is much shorter. This because in the original event log in the period between 04-05-2019 and 26-01-2015 nothing happens except the eleven faulty dated events in the event log. The number of traces and the flow of the event log stay the same.

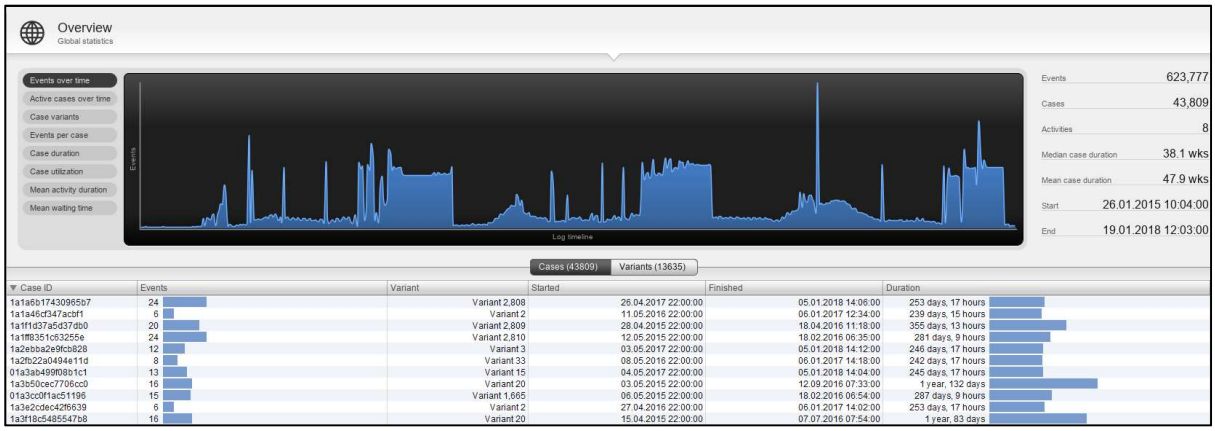

Figure 3: Overview in Disco - Collapsed event log – See in the top right corner the statistics of the event log

4.1.3. Converting CVS to XES

Before the event log can be used in the ProM process mining tool, the different processed CSV files need to be transformed into an XES file. XES stands for eXtensible Event Stream, which is a standard for event logs. The standard was introduced in 2009 by [17], meanwhile the standard has been improved several times and included in the IEEE Standard [18]. XES files are now generally used within the process mining community.

Where for converting CSV to XES until recently separate tools where required, a plug-in in ProM is now available.

Four mapping attributes are especially important for converting the CSV file to a XES file.

- “Case Column” – column where the unique identifier of a trace is specified.
- “Event Column" – column where the events are specified.
- “Start Time” – column where the starting time of an event is specified.
- “Completion Time” – column where the completion time of an event is specified.

It is important to know that case and trace imply the same. For the event log to be usable in the concept drift plug-in *Case Column* and *Event Column* are mandatory. Furthermore, each event needs to have a *Case ID* and a set of events with the same *Case ID* are identified as a trace. Next to this it is essential to use the date format equal to the CSV file to prevent errors in later usage of the XES file.

For the remainder of the choices within the wizard the default settings are kept. Because in this thesis multiple CSV files have been converted to XES and to ensure comparability the input is kept consistent unless stated otherwise.

After successful conversion, ProM presents a dashboard with an overview of the key figures of the created XES file.

Figure 4: Result after converting the CVS file to an XES file - Dashboard view of the XES file in ProM

4.2.Detecting concept drift in a real-life event log

After enhancing the event log and converting it into a XES file the event log can be used ProM. In this chapter we use the concept drift plug-in. The goal is to see if the plug-in is effective on a real-life event log. We need the results of this subsection for further examination of the drift points via process variant analyses.

The results presented in this subsection are created after an iterative process on configuring the concept drift plug-in and altering the event log. For altering the event logs as in subsection 4.1.2 and 4.2.1, we used ideas presented in the paper of [19]. We limit in presenting two results of altered event logs that provide the best results.

4.2.1. Detecting drift using the concept drift plug-in on the enhanced event log

First we want to use the concept drift plug-in on the collapsed event log as described in subsection 4.1. In [5] it is described that a dip in p-value in the concept drift plug-in means a drift point. If a sharp drop occurs from one moment to the next, this can be regarded as a concept drift point. Furthermore, it means that from the concept drift plug-in point of view the process is changing at this point the moment before.

After running the plug-in on the event log created in subsection 4.1 it presents the following result.

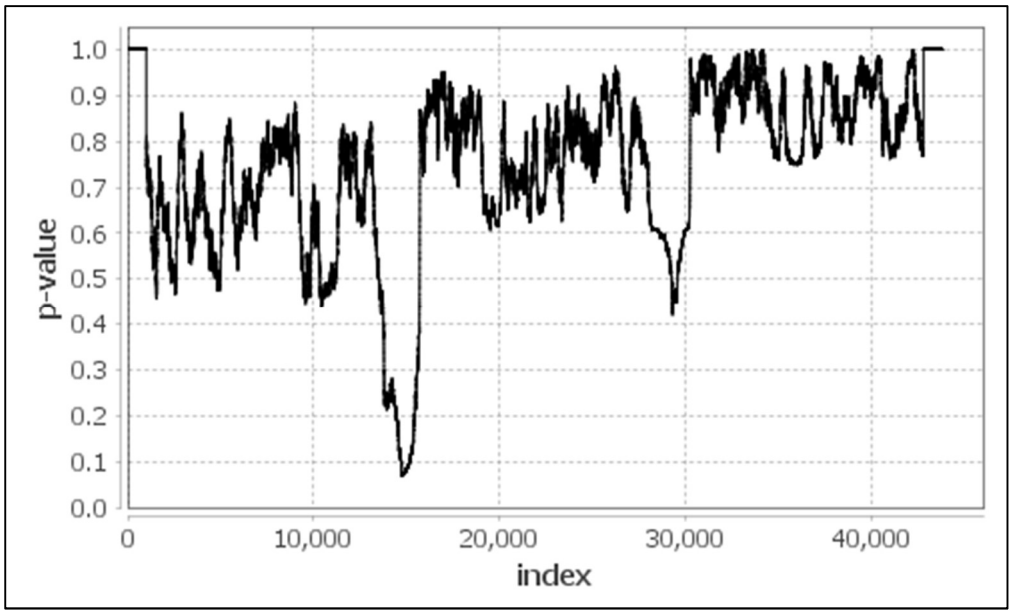

Figure 5: Results Concept Drift Plug-in - Enhanced event log 1

Two drift points are clearly visible. As mentioned earlier the event log comprehends a total of 43.809 traces. Drift points are detected at approximately 14.500 traces, the new inflow of applications of 2016 and 29.000 traces, the new inflow of applications of 2017. Based on this outcome it can be reasoned that this is an annual phenomenon. In subsection 4.3.1 we dive deeper into this happening and attempt to clarify it.

4.2.2. Further enhancing the event log to improve results

Above event log has two shortcomings. There is one *Doctype* that changes over the years but implies the same document. Secondly the event log comprehends more relevant events then just *Doctype* like *Subprocess* and *Activity*. In this subsection we present a solution for these shortcomings and run the enhanced event log trough the concept drift plug-in in ProM to see if we acquire different results.

To improve the flow of the trace over the whole duration of the event log, it may sometimes be necessary to combine events. For example, in the used event log it is indicated in the description of the event log that over the years, for one Doctype three terms are used, “Parcel document”, “Geo Parcel document” and “Department Control Parcels document”. To maintain a better overview over the years, “Geo Parcel Document” has been maintained -and thus replaced if needed- in all years, we have chosen to use “Geo Parcel Document” because it is the only *Doctype* with three *Subprocesses*.

Last major data enhancing technique we want to present is to create a correct and sustainable control flow over the years. For detecting concept drift, we focus on assuring the “Trace-ID”, “Event name”, “Start time” and “End time” are correct and consistent. For later use of the event log it is important that events that are named differently over the years but imply the same are standardized.
A shortcoming of the process illustrated in Table 1is that the process is displayed based on the document flow. This is not entirely in accordance with reality. As can be concluded from Table 4, three document types consist of more than one *Subprocess*. Furthermore, the *Subprocess*

“Application” in *Doctype* "Payment Application" consists of more than one *Activity*. To fabricate a better representation of the real control flow, a combination of *Doctype*, *Subprocess* and *Activity*, column *Event_v2* is added to the event log which offers a better description of the activities.

Doctype	***Subprocess***	***Activity***	**New Column: *Event_v2***
Control Summary	Main	all applicable activities	Update relevant documents
Entitlement application	Change/Objection	all applicable activities	Reopen
	Main	all applicable activities	Calculation of entitlement
Geo parcel documents	Declared/Main	all applicable activities	Record parcel details
	Reported	all applicable activities	Update relevant documents
Inspection	Main	all applicable activities	Inspection
Payment application	Application	abort/begin/finish payment, insert/remove document	Process payments
		mail income, mail valid	Application for payment
		begin editing, calculate, decide, finish editing, initialize, revoke decision, revoke withdrawal, withdraw	Finalize decision
	Change/Main/Objection	all applicable activities	Reopen
Reference alignment	Main	all applicable activities	Validate Parcel information

Table 4: Adding an extra column Event_v2 which better reflects the real process

For the new enhanced event log the CSV file is again converted to XES. The main difference in the conversion process with regard to subsection 4.2.1 is that in this case *Doctype* is not chosen as “Event type” but of our newly created class *Event_v2*. This provides the following results as presented in Figure 6.

As can be clearly seen, the pattern of the results from subsection 4.2.1 and performed in this subsection is largely the same. The main drift points again appear at trace 14.500 and trace 29.000.

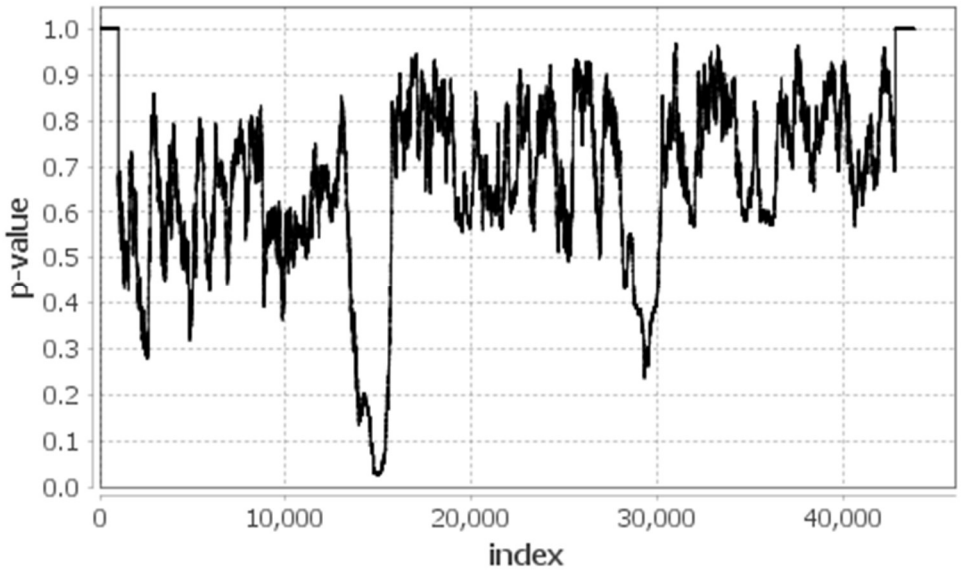

Figure 6: Results Concept Drift Plug-in - Enhanced event log 2

We hoped that further customizing and improving of the event log would yield different results. However, the drift points stay the same. Although they are more visible in the second experiment. However, it has been demonstrated that drift points can be demonstrated in real-life event logs, if handled appropriately. A condition for this technique is that real-life event logs which are often large, and complex must be enhanced before they can be used.

4.2.3. Checking for drift points in the different years

Because it is clear that the drift points occur during the transition of a year to the following year. We are curious to see whether we can discover drift points in a specific year. To do this we split the event log in three parts taking out the transition point from year to year. The event log created in sub section 4.2.2 is divided into three parts, one for each year in which a case is started. The distribution of events and traces in these split event logs is illustrated in Table 5.

Year	Events	Traces	E/T
2015	311.230	15.137	20,56
2016	165.389	14.959	11,06
2017	209.101	14.887	14,05
Totals	685.720	44.983	15,24

Table 5: Distribution Events and Traces over the Years including average events per trace (E/T)

After performing the same procedures as performed in the aforementioned sub-sections, we acquire the following results presented in Figure 7, Figure 8 and Figure 9. Although there are no clear drift points within the years, we can safely conclude the following. The volatility of the graph decreases over the years. Wherein 2015 we see values ranging from 0.30 to 0.85, 2017 presents a more stable pattern with values ranging from 0.60 and 0.95. So not only there is a lower spread also in general the values are higher. This could mean that cases are executed with fewer events or a lower variation of events.

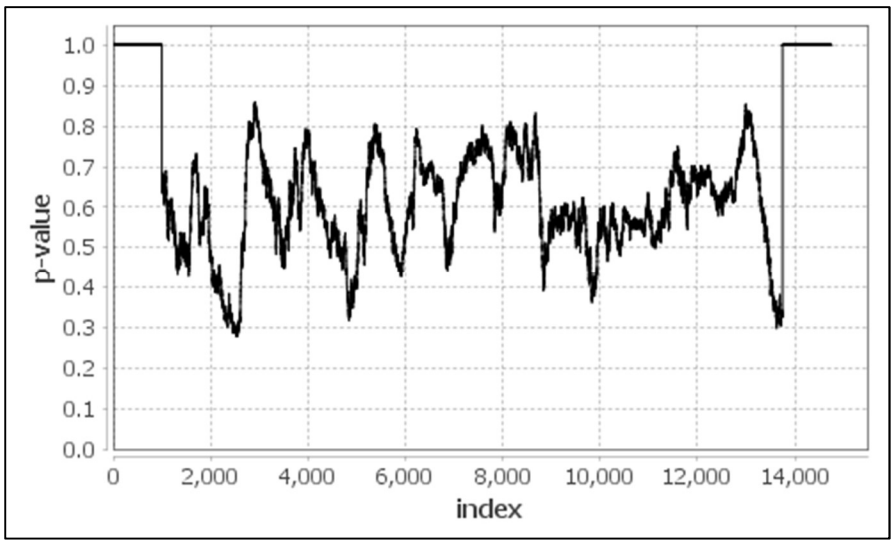

Figure 7: Results Concept Drift Plug-in - Enhanced event log 2015 – p-value between 0.30 and 0.85

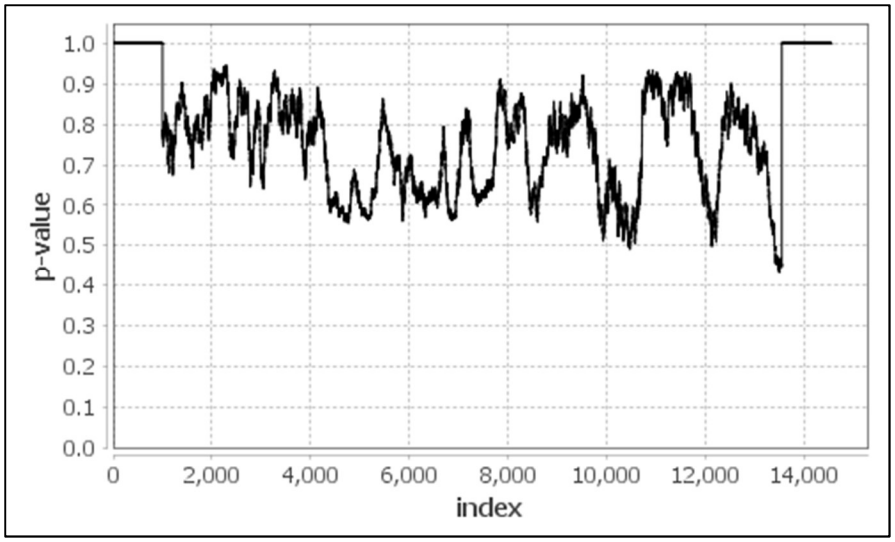

Figure 8: Results Concept Drift Plug-in - Enhanced event log 2016 – p-value between 0.45 and 0.95

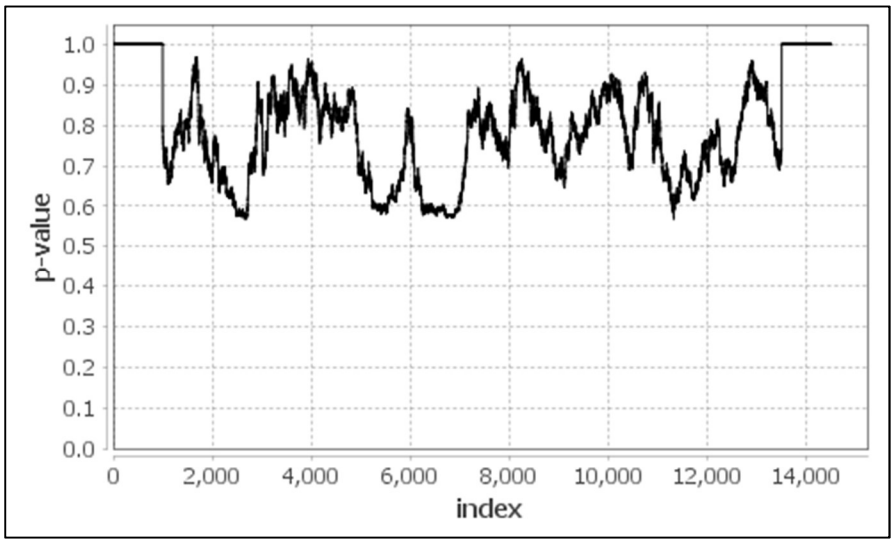

Figure 9: Results Concept Drift Plug-in - Enhanced event log 2017 – p-value between 0.60 and 0.95

The results in this subsection are surprising and unexpected. Although there are no real drift points detected but it is evident that the process evolves over the years. It seems that this results in a process that is more stable and efficient. Summarizing, it looks like that the process drift plug-in can not only be used to detect concept drift points. It can also, if a large enough dataset is available, determine how an event log evolves in terms of stability and efficiency. In the next chapter we investigate whether we can analyze a difference in the number of variances over departments.

4.2.4. Comparing the concept drift plug-in results between departments

We find it interesting to see that if we use the method from subsection 4.2.3 a volatility comparison between departments is possible.

In the event log there are four departments recorded. For this analysis we made a further split. For each year and per department we have made a separate event log. A total of twelve new event logs are made. The distribution of events and traces in these further split event logs is illustrated in Table 6. Based on this simple overview department 4e has the least events per trace on average. This already hints that department 4e works more efficiently than other departments.

Year	4e			6b		
	Events	Traces	E/T	Events	Traces	E/T
2015	84.599	4.580	18,47	84.888	4.181	20,30
2016	47.404	4.531	10,46	43.377	4.139	10,48
2017	58.442	4.528	12,91	62.180	4.111	15,13
Totals	190.445	13.639	13,96	190.445	12.431	15,32
Year	d4			e7		
	Events	Traces	E/T	Events	Traces	E/T
2015	44.750	1.930	23,19	96.993	4.446	21,82
2016	24.440	1.910	12,80	50.168	4.379	11,46
2017	30.088	1.904	15,80	58.391	4.344	13,44
Totals	99.278	5.744	17,28	205.552	13.169	15,61

Table 6: Distribution Events and Traces over the years and departments including average events per trace (E/T)

Now we want to dive deeper into the different newly created event logs. Running the event log through the concept drift plug-in in ProM gives the following drift plots presented in the below figures.

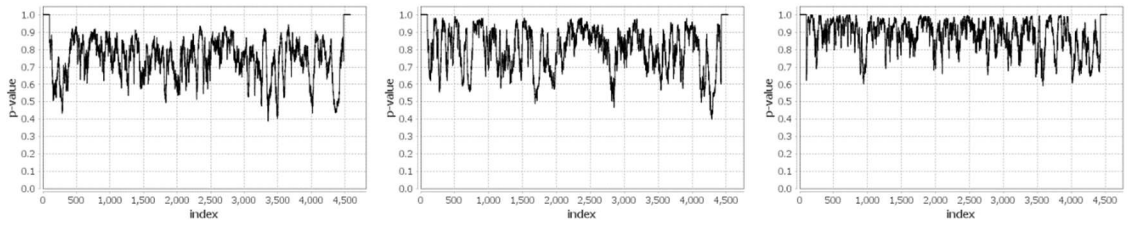

Figure 10: Results of the concept drift plug-in for department 4e – from left to right 2016, 2016 and 2017

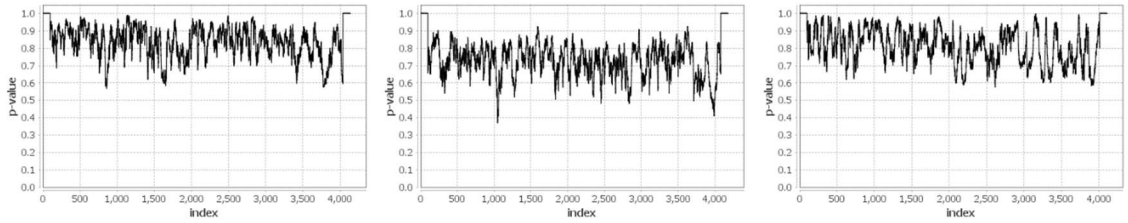

Figure 11: Results of the concept drift plug-in for department 6b – from left to right 2016, 2016 and 2017

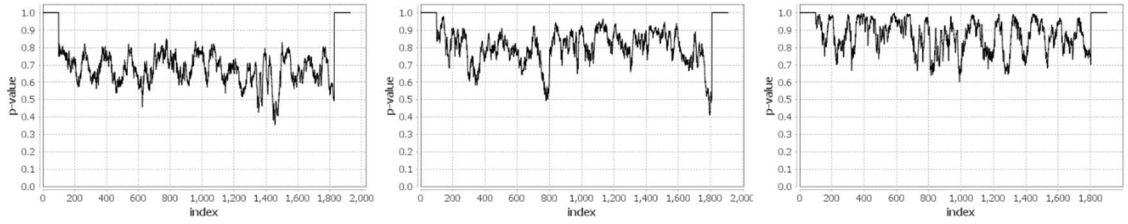

Figure 12: Results of the concept drift plug-in for department d4 – from left to right 2016, 2016 and 2017

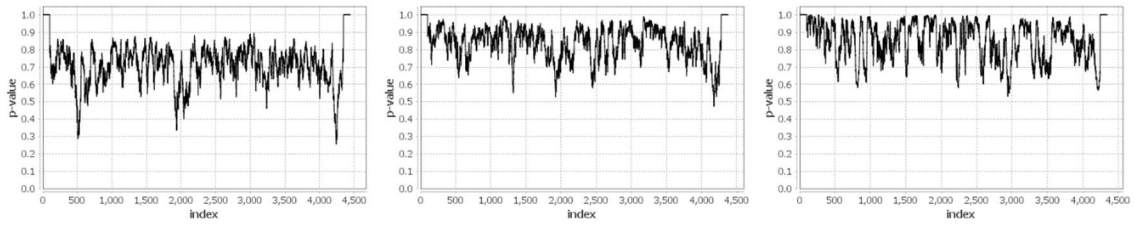

Figure 13: Results of the concept drift plug-in for department e7 – from left to right 2016, 2016 and 2017

In this experiment the results are closer together which makes it difficult to draw direct conclusions. Although the spread in department 4e in 2017 seems to be the least and the spread in department e7 in 2015 is the greatest. In chapter 4.3 we use variant analyses to see if we can further explain the differences between departments.

In general, an analysis between departments on volatility can be performed to determine which department works more stable and to see if best practices can be established that can be implemented in the other departments.

4.3.Validating the results from the concept drift plug-in

In the previous chapter we concluded there are drift points where one grant year transfers to the next grant year. Secondly, we have seen that over the years the process has become less volatile. We are interested in evaluating if there were significant differences between the departments. It is possible to use the concept drift on split event logs which only cover certain departments. In this chapter we make an effort to explain these observations by further analyzing the event log and examining the trace variations. For this we use the variation filter in Disco. The presented approach is a practical solution by comparing the different results from subsection 4.2 in a variant analyses.

4.3.1. Explaining the drift points in between grant years

First, we examined the enhanced event log. Regarding concept drift points detected in 4.2.1 it is striking that in both 2016 and 2017 there is a sharp drop in the number of applications at about the same points in the event log. The active cases a day drop from 6.762 on 18-02-2016 to 3.315 on 20-02-2016. About the same numbers are visible in January 2017. Then in the beginning of May there is a sharp increase. We have cross checked this with the German legislation regarding the deadline of applying for the direct payments [20]. On the subsidy providers website, the ultimate delivery date for an application is 15 May. This decrease in number of traces clearly corresponds to the drift points presented in subsections 4.2.1 and 4.2.2.

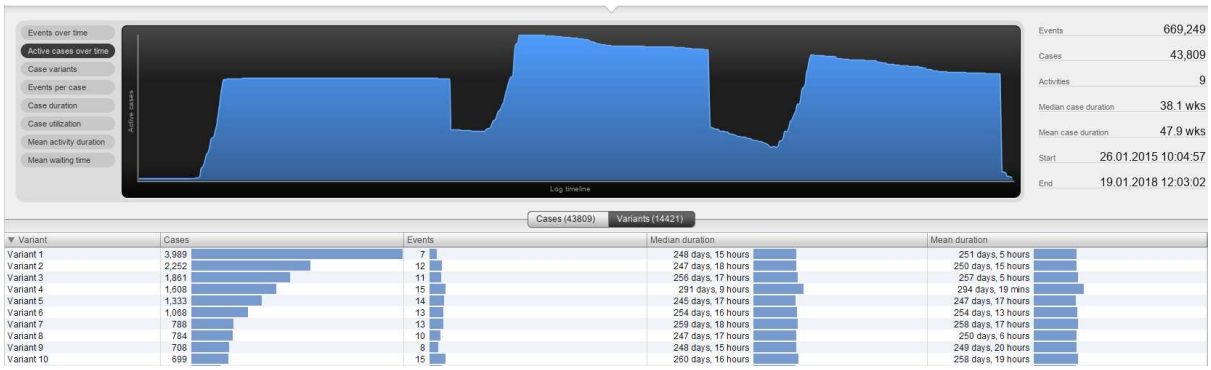

Figure 14: A visual representation of the fallback of active traces and an overview of the first ten variants with the number of traces that comply to that variant

4.3.2. Explaining volatility changes over the years

Further exploration of the log presents a high variation of traces. There are 14.421 different variants of the traces detected. A variant differs if the events are in a different order, particular events do or do not happen, the number of events deviates or a combination. The largest variant applies to 3.989 traces. There are 12.627 variants with just one trace. The largest ten variant contain 15.090 traces, totaling 34% of the traces in the whole event log.

We try by means of the variant analysis to explain and verify the decrease in volatility over the years observed in sub section 4.2.3. It can be stated that variant with the highest number of traces has the highest consistency, we first focus on the ten largest variants. With this selection we cover 34% of all traces and 24% of all events. An analysis on these ten variants reveals that 58% of these traces happen in 2017. In 2015 there is least consistency in variance with only 10% of the cases. This verifies the results of sub section 4.2.3 that over the years the way of working has become more consistent.

Figure 15: Distribution of the 10 most common variants over time

To put it to the test, we have created a filter for the variants with the least cases. For this we selected the variants with between one and four traces. This covers 96% of all variants and 35% of all traces. The distribution clearly illustrates that the inconsistent processes are mainly in 2015 and 2016 as visualized in Figure 16. This further strengthens the observation that volatility of the execution of the traces has decreased over the years.

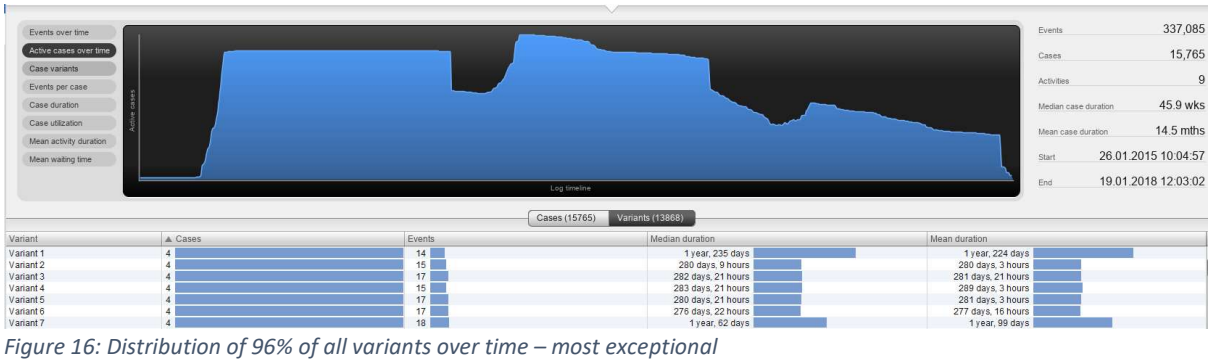

Figure 16: Distribution of 96% of all variants over time – most exceptional

4.3.3. Explaining different volatility levels between departments

In subsection 4.2.4 we have presented the different volatility levels between departments. Because the results were fairly close to each other, it was difficult to draw a firm conclusion. In this subsection we want examine the created event logs through a variant analysis. Table 7 shows the number of traces and the number of variations per year per department. Dividing traces by variants (T/V) gives the average number of traces per department. With the overview of Table 7 it is easy to

compare the departments and see where and when the least variations with respect to the traces occur.

Our suspicion that the lowest volatility takes place in 2017 by department 4e is confirmed. There is a sharp drop of variants while the executed traces stay the same. This further confirms our observation in subsection 4.3.2 that in 2017 there is -in general- a lower volatility across the departments. We see a clear trend where in 2017 there is less variation in the traces. This means that the work was performed in a more standardized manner in 2017

Year	4e			6b		
	Traces	Variants	T/V	Traces	Variants	T/V
2015	4.580	2.223	2,06	4.181	2.513	1,66
2016	4.531	1.142	3,97	4.139	1.115	3,71
2017	4.528	538	**8,42**	4.111	714	5,76
Totals	13.639	3.903	3,49	12.431	4.342	2,86
Year	d4			4e		
	Traces	Variants	T/V	Traces	Variants	T/V
2015	1.930	1.472	1,31	4.446	3.018	1,47
2016	1.910	927	2,06	4.379	1.502	2,92
2017	1.904	411	4,63	4.344	764	5,69
Totals	5.744	2.810	2,04	13.169	5.284	2,49

Table 7: Overview of the number of traces and variants per year and per department showing when and where on average the least variation occurs – T/V means traces divided by variants

5. Discussion

In this chapter we start with a discussion on the research. After this we present our conclusion. Furthermore, we want to present practical recommendations how this research can be used in a practical way within organizations. We have views of future research that comes out of this research. We want to conclude this chapter with a reflection on the quality and maintainability of this thesis.

5.1. Discussion on the research

In this research we have combined the validation of a statistical method in the form of the concept drift plug-in on a real-life event log with further examining the results from the concept drift plug-in in a practical manner trough variant analysis. A valuable by-product of the iterative process is an easy to use and firsthand approach for comparing years and departments to distinguish when or where the executed process is less volatile.

5.1.1. Validating the concept drift plug-in with a real-life event log

One of the goals of this research is to validate the concept drift plug-in presented by [5]. In the research we have been able to demonstrate that the plug-in works. However, in such large and complex data sets as used in this thesis it is difficult to validate whether all drift points have been found. Especially when business context is missing. In this event log, drift points were found during the transition of an application year. We were able to demonstrate that just before the deadline of application, which is available on the website of the subsidy provider, there is a substantial influx of applications. Exactly when the concept drift plug-in presents a significant dip in the plot from the plug-in in ProM. Because the event log spans three years, we were able to demonstrate it for two consecutive years. The shortcoming of this analysis is that it is quite logical that the plug-in responds

to such a decrease in the number of applications. Basically, we have revealed the plug-in can be used to detect a significant increase or decrease in the number of new traces. The question is whether this intention of the plug-in and has added business value. Therefore, it remains a big challenge to prove that all the drift points have been discovered, we have experimented further with different configurations.

5.1.2. Using the concept drift plug-in for volatility analysis

During the iterative process of using the concept drift plug-in we discovered an unplanned insight we want to present. One approach was to extract drift points from the event log which was limited to one application year. We did this by splitting the event log per application period of one year. After this we ran the three newly made event logs through the concept drift plug-in in ProM again. Although there were no clear drift points detected it gave interesting results. Over the years the spread in the concept drift plot lessens. Values are higher in a general sense as clearly presented in subsection 4.2.3. In general, the drift plots from ProM present a less volatile representation in the application period of 2017 than the application period of 2015. It appears that the process took place in 2017 with less variation. The numbers say that the traces with the least events happen in 2016. From this we deduce that the volatility depends not only on the number of events per trace, but also on the variation in the order of events and whether events are present.

In summary, we were able to present an approach which was not planned. It was clear that the drift plot of 2017 presented in Figure 9 has a lower spread then the other years presented. We have recognized this as a year where the process is executed in a less volatile way. Meaning less variations in the traces, or so to say a more stable process.

5.1.3. Examining the concept drift plug-in results through variant analyses

In short, the variant analysis confirms our observations from the results of the concept drift plug-in.

The concept drift points are explained through variant analyses. On the concept drift points we have seen a sharp increase of applications. It is however questionable what the added value is. We have not been able to find clear drift points that the process intrinsically changed in this event log. Based on this event log we could not find other sudden drift points as described in subsection 2.2 then the influx of applications. This is unfortunate. It can mean that in real-life event logs the process change is does not reveal itself suddenly but develops slowly. It can also mean that there are drift points in this event logs, but the concept drift plug-in has not found them. This reveals a general weak point of this research, there is no business context. In other words, we cannot find out in any other way whether there are drift points.

Detecting volatility is an unexpected insight for which the concept drift plug-in seems useful. The variant analysis shows that from the traces in the top 10 variants 58% happen in 2017. We have also demonstrated that the largest variation of traces happens in 2015 and 2016. Variation analysis is an adequate approach to validate the volatility results from the concept drift plug-in.

5.2.Conclusion

The problem statement of this thesis is:

How and to what extent can process variant analyses be used to examine concept drift from a control flow perspective?

We have shown that with the correct preparation of the event log drift detection is possible for a real-life event log. We must acknowledge that we have only been able to find drift points where a sharp influx of applications takes place. We were unable to find drift points for intrinsic process changes. A weak point is that we could not investigate whether false negatives are overlooked, because we did not have enough knowledge of the business context.

Furthermore, with the same tooling, we have observed a decrease of volatility over the different grant years. For further examination of the approach, the same volatility analysis has been performed on the different departments. In further practical experiments we have presented approaches to validate these hypotheses, supplemented in one instance with information from the regulation.

Especially the volatility analysis with the concept drift plug-in and examining it with variant analysis is a new and unexpected approach which comes out of this research. The approach is a tool that can be used by businesses to enhance the processes. Based on the outcomes of the volatility analysis certain periods or departments can be further examined to improve the way of working, and so, hopefully accomplish a business advantage.

5.3.Practical recommendations

For organizations that want steady processes in their day to day work an analysis of the event logs can have added value. Using the approach presented in this thesis an organization can detect concept drift points but can also use the same ProM plug-in to compare volatility between years or departments. This meaning that one can pinpoint periods or places where there is a difference in the stability of the process. Further investigation in these periods can help improve the processes. It can make the process more stable, faster, and cheaper.

This technique can be applied generically. For example, certain time periods can be compared, but it can also be determined whether one department has a less volatile process than the other department. This is especially useful when the departments work with the same process. A deep dive on the process can be performed on the time periods or departments in which the process is least volatile. From this deep dive best ways of working can be derived which can be implemented in general or in the other departments which seem to work less efficiently.

A shortcoming of this approach is that business user unfriendly tools must be used. Especially the ProM tool is not practical for business users due to being a complex researching tool. Furthermore, preparing a real-life event log is difficult and time consuming.

5.4.Future Work

For future work we advise to see if the detected concept drift can be further explained. In this thesis it is clear that the results from the concept drift broadly correspondent with the variant analysis. Future work could further specify reason for the drift. For this thesis we have used a real-life event log but lacked business context. When this approach is applied to an event log where the business context is clear, the drift may be better explained.

It is striking to see the substantial work that goes in preparing a real-life event log before it can be used for the approach discussed in this thesis. Future work could add off the shelf techniques to manage an event log for concept drift or variant analyses. Especially in making it lightweight and comprehensible.

5.5.Reflection

With this thesis we try to add a pebble to the scientific spectrum and specifically the business process management domain. In this section we want to present a reflection on the performed research. We want to share our thoughts on the quality of the research and want to see to what extent the conclusions are tenable.

5.5.1. Strong points

We find that one of the strong points is the method of bringing together a statistical technique, the concept drift plug-in with a practical method in the form of the variant analysis. By validating the results of the concept drift plug-in in another tool we intend to further strengthen the designed approach.

By working in an iterative way, we have found a new insight of usage the concept drift plug-in which adds value to the process mining community.

5.5.2. Weak points

One of the weak points of the thesis is that we could not present a technique to identify false negatives. In a large and complex, real-life event log used it is hard to assure that the concept drift plug-in has found all the concept drift points. Checking the event log for drift points that have been overlooked by manually seem an impossible or at least a very time-consuming task.

Another identified weakness is the lack on business context of the event log. Which limits triangulation because we could not verify with the process executioners. Although limited information was provided with the event log consultation was not possible. For instance, we wonder if the years supplied in the event log are complete years. In other words, are all the applications, thus traces, fully completed or are there incomplete traces in the event log. This could have an impact and the results presented in the paper.

6. References

1. Van Der Aalst, W.M.P., A.H.M. Ter Hofstede, and M. Weske. *Business process management: A survey*. in *International conference on business process management*. 2003. Springer.
2. Smit, K., M. Zoet, and J. Versendaal, *Identifying Challenges in Business Rules Management Implementations regarding the Elicitation, Design, and Specification Capabilities at Dutch Governmental Institutions* Journal of Information Technology Theory and Application 2018. **19**(2): p. 121-137.
3. Boyer, J. and H. Mili, *Agile Business Rule Development*. 2011.
4. the Business Rules Group, *Defining Business Rules ~ What Are They Really?* 2000.
5. Bose, R.P.J.C., et al. *Handling Concept Drift in Process Mining*. 2011. Berlin, Heidelberg: Springer Berlin Heidelberg.
6. Carmona, J. and R. Gavalda. *Online techniques for dealing with concept drift in process mining*. in *International Symposium on Intelligent Data Analysis*. 2012. Springer.
7. Bolt, A., M. de Leoni, and W.M.P. van der Aalst, *Process variant comparison: Using event logs to detect differences in behavior and business rules.* Information Systems, 2018. **74**(1): p. 53-66.
8. Polpinij, J., A. Ghose, and H.K. Dam, *Mining business rules from business process model repositories.* 2015.
9. Ye, J., et al., *Ontology-based models in pervasive computing systems.* The Knowledge Engineering Review, 2007. **22**(4): p. 315-347.
10. Maaradji, A., et al., *Fast en Accurate Business Process Drift Detection.* 2015.
11. Bose, R.P.J.C., et al., *Dealing With Concept Drifts in Process Mining.* 2013.
12. Fluxicon. *Disco*. Available from: https://fluxicon.com/disco/.
13. University, P.M.G.-E.T. *ProM 6.9*. Available from: http://www.promtools.org/doku.php?id=prom69.
14. Smyth, P.G., R.M., *Rule Induction Using Information.* Knowledge Discovery in Databases, 1999: p. 159-176.
15. Commission, E. *Financing the common agricultural policy*. Available from: https://ec.europa.eu/agriculture/cap-funding_en.
16. 2018, t.I.W.o.B.P.I. *Business Process Intelligence Challenge 2018*. Available from: https://www.win.tue.nl/bpi/doku.php?id=2018:challenge.
17. Gunther, C., *XES Standard Edition*. 2009: http://www.xes-standard.org/_media/xes/xes_standard_proposal.pdf.
18. Engineers, I.o.E.a.E., *IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams.*
19. Wangikar, L.D., Sumit; Yadav, Abhilasha; Bikshit, Bhavy; Yadav, Dikshant;, *Faster Payments to Farmers: Analyses of the Direct Payments Process of EU's Agriculture Guarantee Fund.* 2018.
20. Baden-Wurttemberg, M.f.L.R.u.V. *Direktzahlungen und Ausgleichsleistungen für landwirtschaftliche Betriebe*. Available from: https://foerderung.landwirtschaft-bw.de/pb/MLR.Foerderung,Lde/Startseite/Foerderwegweiser/Betriebspraemie.